1

Teletraffic Theory and Engineering

Robert B. Cooper and Daniel P. Heyman

I. INTRODUCTION

A telecommunications network consists of expensive hardware (trunks, switches, etc.) with the function of carrying telecommunications traffic (phone calls, data packets, etc.). The physical network is fixed, but the traffic that it is designed to carry is random. That is, the times at which calls are generated are unpredictable (except in a statistical sense), and, similarly, the lengths of time that the calls will last are unpredictable; yet, the network designers must decide how many resources to provide to accommodate this random demand. If the resources are provided too sparingly, then the quality of service will be low (e.g., too many calls will be lost because the required resources are not available when needed); but, if the resources are provided too generously, then the costs will be too high. *Teletraffic theory* deals with the mathematical analysis of models of telecommunications systems and with the interrelationships among the provision of resources, the random demand, and the quality of service; *teletraffic engineering* addresses the art and science of the application of this theory to the design of real telecommunications systems.

Encyclopedia of Telecommunications, Volume 16, Pages 453–483
Copyright © 1998 by Marcel Dekker, Inc., 270 Madison Ave., New York, New York
All rights reserved.

Publisher and Contributor hereby agree that the Entry has been specially commissioned by the Publisher for use as a contribution to a collective work and shall be and is considered a ‘‘work-made-for-hire.’’ Publisher shall own all right, title and interest in and to the Entry, including without limitation the entire copyright therein, and shall be deemed to be the author of the Entry for copyright purposes. Publisher shall be free to use the Entry, in whole or in part, in all languages, throughout the world, in perpetuity, in any form or medium now known or hereafter developed, and to license others to do the foregoing.

II. HISTORY

The telephone was patented in 1876, and the first commercial telephone switchboard went into operation in 1878 in New Haven, Connecticut. It consisted of a set of subscribers who could be connected two at a time via a single path. It has been said that the need for teletraffic theory became apparent as soon as the number of subscribers grew to three! The first significant advance in teletraffic theory came in 1917, when A. K. Erlang, a scientist/mathematician/engineer working for the Copenhagen Telephone Company, published a paper that described a method and used it to derive some formulas that provide the basis for much of modern teletraffic theory and engineering.

Later, with the invention of operations research during World War II, Erlang's methods and models were incorporated into queueing theory, and these two subjects (queueing, teletraffic theory) are now closely intertwined. (A *queue* is a waiting line. *Queueing theory* is the mathematical theory of systems that provide service to customers with arrival times and service requirements that are random. If servers are unavailable to accommodate arriving customers, then a queue may form, hence the name.) There is now a huge amount of literature on queueing (and teletraffic) theory, and papers are being published in the technical journals at an ever-increasing rate.

In this article, we survey basic teletraffic (and queueing) theory, and we discuss both classical applications and new theory for applications that are driven by recent advances in telecommunications technology and computer science.

III. BASIC CONCEPTS

We take as our basic model a system in which calls arrive at random times, and each call requests the use of a *trunk*. (In this article, we will ignore the distinction between trunks, which interconnect the switches, and lines, which connect the subscribers to the switches.) If a trunk is available, the call holds it for a random *holding time*, and if no trunk is available, the *blocked call* takes some specified action, such as overflowing, retrying, or waiting in a queue. (In queueing theory parlance, the calls are *customers*, the trunks are *servers*, and the holding time is the *service time*.) The objective of teletraffic theory is to derive appropriate descriptions of the random teletraffic (a description of the statistical, or *stochastic*, properties of the arrival times and holding times) and to derive formulas that describe the performance of the system (e.g., the probability of blocking, the fraction of calls that overflow, the average waiting time, etc.) as a function of the demand and the number of trunks. This theory is then adapted and applied to the design and administration of real telecommunications systems; that is teletraffic engineering.

The central concept of teletraffic engineering is the stochastic nature of teletraffic, so the underlying mathematics used are *probability*, *statistics*, and *stochastic processes*. Therefore, we summarize (as briefly as possible) these mathematical processes. Then, we apply this to derive and understand the basic formulas of teletraffic theory. To make this theory concrete and to explore the robustness of these formulas, we describe briefly the essential concepts of simulation, and we give pseudocode that can be used to write computer programs to simulate these models.

A. Birth-and-Death Process

To fix these concepts, consider the following model. Calls arrive according to a stochastic process (described below) at a group of s identical trunks. If an arriving call finds a trunk available, the call holds the trunk for a random holding time (described below), after which the call drops the trunk, which then becomes available for another call. If the arriving call finds all s trunks busy, then the call is *blocked*, in which case it takes some specified action (described below).

Suppose that P_j denotes the long-run probability that the system is in state j, that is, P_j is the probability that the number of calls present (in service or, if the model permits, waiting in the queue for a trunk to become available) is j; assume that when the system is in state j, then the call arrival rate is λ_j, and the call departure rate is μ_j. Then, it can be shown that, under certain conditions that must be satisfied by the arrival process and the departure process (discussed below), the following equations determine the state probabilities as a function of the rates λ_j and μ_j:

$$\lambda_j P_j = \mu_{j+1} P_{j+1} \qquad (j = 0, 1, 2, \ldots) \tag{1}$$

and

$$P_0 + P_1 + \cdots = 1 \tag{2}$$

Equation (1), originally derived by Erlang, can be given the following interpretation: rate up from state j equals rate down from state $j + 1$. That is, the term $\lambda_j P_j$ on the left-hand side of Eq. (1) equals the fraction of time P_j that there are j calls present multiplied by the rate at which calls arrive when there are j calls present; hence, the product $\lambda_j P_j$ equals the long-run rate (in transitions per unit time) at which the system state jumps from level j to level $j + 1$. Similarly, the right-hand term $\mu_{j+1} P_{j+1}$ equals the long-run rate (in transitions per unit time) at which the system state jumps down from level $j + 1$ to level j. Therefore, if the system is to be in "equilibrium," Eq. (1) must hold. Successive solution of Eq. (1) for each P_j in terms of the previous ones gives

$$P_j = \frac{\lambda_0 \lambda_1 \cdots \lambda_{j-1}}{\mu_1 \mu_2 \cdots \mu_j} P_0 \tag{3}$$

and P_0 is calculated from the normalization condition, Eq. (2) (which simply requires the sum of the fractions of time that the system spends in each state to add to 100%).

$$P_0 = \frac{1}{1 + \frac{\lambda_0}{\mu_1} \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} + \cdots} \tag{4}$$

A stochastic process that is described by Eq. (1) is called a *birth-and-death process*. The key technical point here is that the instantaneous rates λ_j and μ_j are assumed to depend on only the present state and are otherwise independent of the past history of the process. The birth-and-death probabilities P_j ($j = 0, 1, \ldots$) defined by Eq. (1) are *time-average probabilities*; that is, P_j can be interpreted as the fraction of time that the system spends in state j. Also of interest are the *customer-average probabilities* Π_j ($j = 0, 1, \ldots$); Π_j can be interpreted as the fraction of customers arriving when the system is in state j. In general, the fraction of time that the system spends in a given state does *not* equal the fraction of customers finding that state when they arrive. However, when the customers arrive according to a *Poisson process* (defined below), then

$$\Pi_j = P_j \qquad (j = 0, 1, \ldots) \tag{5}$$

The important equality Eq. (5) reflects the *PASTA theorem* (Poisson Arrivals See Time Averages). Sometimes, the P's are called the *outside observer's distribution* (reflecting the notion that they measure the frequencies of occurrence of the states as seen by an outside observer passively observing the system continuously or at random instants), and the Π's are called the *arriving customer's distribution* (reflecting the notion that they measure the frequencies of occurrence of the states as seen by the arriving customers). The PASTA theorem says that, remarkably, a stream of Poisson arrivals will see the states with the same frequencies as will an outside observer, even though the arrivals, in general, "cause" the states of the system and view the system just prior to the instants of upward state transitions, whereas the outside observer has no causal effect on the states of the system. (There are some situations for which non-Poisson arrivals see time averages, but these are rather special.) We now discuss how these results are applied to our basic teletraffic model.

B. Poisson Process

The usual assumption in classical teletraffic theory (and the assumption that is most reasonable in the absence of evidence to the contrary) is that the call arrivals follow a Poisson process. It turns out, as the following physical argument would suggest, that the Poisson assumption is consistent with data for voice traffic when the calls are generated by a large number of independently acting subscribers.

Assume that time is divided into equal-length intervals of length Δt and (1) there can be at most one arrival in each interval, (2) the probability of an arrival in any given interval is proportional to the length Δt, and (3) the intervals are statistically independent of each other.

Let the random variable X be the length of time from now (Time 0) until the arrival time of the next call. We calculate the probability $P(X > t)$ that no call arrives in the interval $(0, t)$. Imagine that $(0, t)$ is divided into n intervals, each of length $\Delta t = t/n$. If we denote by λ the proportionality constant assumed in Item 2 above, then the probability that an arrival occurs in any given interval of length Δt is $\lambda \Delta t = \lambda t/n$, and, hence, by Item 3, the probability of no arrivals in any of the n Δt's that comprise the interval $(0, t)$ is $(1 - \lambda t/n)^n$. We now pass from discrete time to continuous time by imagining that $\Delta t \to 0$ or, equivalently, $n \to \infty$. That is,

$$P(X > t) = \lim_{n \to \infty} \left(1 - \frac{\lambda t}{n}\right)^n \tag{6}$$

It is well known (see any calculus text) that the limit on the right-hand side of Eq. (6) equals $e^{-\lambda t}$; hence, if we let $F_X(t) \equiv P(X \leq t)$ denote the distribution function of X, then Eq. (6) becomes

$$F_X(t) = 1 - e^{-\lambda t} \qquad t \geq 0 \tag{7}$$

A random variable with a distribution function given by Eq. (7) is said to be *exponentially distributed*, and the process that describes arrivals with interarrival times that are iid (independent and identically distributed) with the distribution function Eq. (7) is said to be a Poisson process. Thus, if the call arrival process can be described by Items 1–3 (and what could be simpler and still make sense?), then the calls arrive according to a Poisson process.

If we let $E(X)$ be the expected value of X, that is,

$$E(X) = \int_0^\infty t f_X(t)dt \tag{8}$$

where $f_X(t) = d/dtF_X(t)$ is the *density function* of X (see any probability text), then substitution of Eq. (7) into Eq. (8) yields

$$E(X) = 1/\lambda \tag{9}$$

that is, $\lambda = 1/E(X)$, and hence the proportionality constant posited in Item 2 is the long-run arrival rate.

An important property of any random variable that is exponentially distributed [that is, with a distribution function given by the right-hand side of Eq. (7)] is the *memoryless* (or *Markov*) *property*, which can be expressed as follows: for all $y \geq 0$ and $t \geq 0$,

$$P(X > y + t \mid X > y) = P(X > t) \tag{10}$$

Equation (10) says that the conditional probability that an exponential variable X lasts longer than $y + t$ if it is known to have lasted longer than y (that is, "given" $X > y$) *does not depend on the value of* y. [Equation (10) is easily proved using Eq. (7) and the familiar definition of the conditional probability of occurrence of an event given the occurrence of any other event.]. It can be shown that Eq. (7) is the only continuous distribution that satisfies Eq. (10); thus, Eqs. (7) and (10) are equivalent characterizations of the exponential distribution.

In the context of the birth-and-death process, described by Eq. (1), we see that if the call arrivals follow a Poisson process with rate λ, then the instantaneous birth rate λ_j when the system is in the state j is the same for all states, and we can take $\lambda_j = \lambda$.

Let us now assume that the holding times are iid exponential random variables, with average length τ, say. This is a much less "reasonable" assumption for holding times than for interarrival times, because the Markov property Eq. (10) seems questionable if the random variable X is taken to represent the length of a call (rather than the time separating a pair of arriving calls). But, it is precisely this property of memorylessness that permits the application of Eq. (1); so, we assume "exponential holding times" for expediency. (This modeling assumption will turn out to be much better than might appear at this point in the discussion.) Thus, if S represents a generic holding time, and the average holding time is denoted by $E(S) = \tau$, then the exponential-holding-times assumption implies that $F_s(t)$ is given by the right-hand side of Eq. (7), with the rate λ replaced by $\mu = 1/\tau$ (μ is the *service rate*).

Another easily verified property of the exponential distribution is that the minimum of a set of independent exponentially distributed variables is also exponentially distributed, with a rate equal to the sum of the original rates. Thus, if there are k iid exponential calls in progress simultaneously, then the time until the shortest of them ends is exponential with rate $k\mu$, where μ is the individual service rate; that is, the aggregate instantaneous call completion rate is $k\mu$.

IV. THE ERLANG B, ERLANG C, AND ENGSET MODELS

If the arrival process is Poisson with rate λ and the holding times are exponential with average length $1/\mu$, then the state probabilities P_j are determined by the birth-and-death Eq. (1), with $\lambda_j = \lambda$ and $\mu_j = j\mu$ when $j \le s$ and $\mu_j = s\mu$ when $j > s$. The solution to Eq. (3) for $j \le s$ is

$$P_j = \frac{(\lambda/\mu)^j}{j!} P_0 \qquad (j = 0, 1, \ldots, s) \tag{11}$$

A. Erlang B Model

If we assume that *Blocked Calls are Cleared* (BCC), then obviously $P_j = 0$ for all $j > s$, and Eq. (2) gives

$$P_0 = \frac{1}{\sum_{k=0}^{s} \frac{(\lambda/\mu)^k}{k!}} \tag{12}$$

If we let

$$a = \lambda/\mu \tag{13}$$

then Eqs. (11) and (12) can be written

$$P_j = \frac{\frac{a^j}{j!}}{\sum_{k=0}^{s} \frac{a^k}{k!}} \quad (j = 0, 1, \ldots, s) \tag{14}$$

The set of probabilities defined in Eq. (14) is called the *Erlang loss distribution* (derived by Erlang in 1917). In particular, the probability that all trunks are busy is denoted by $P_s \equiv B(s, a)$, the well-known *Erlang B* or *Erlang loss formula*:

$$B(s, a) = \frac{\frac{a^s}{s!}}{\sum_{k=0}^{s} \frac{a^k}{k!}} \tag{15}$$

The Erlang B formula is sometimes called *Erlang's first formula*, denoted by $E_{1,s}(a)$, so $E_{1,s}(a) \equiv B(s, a)$. We now address some of the ramifications and interpretations of Eq. (15) and then briefly discuss related models, such as Erlang C and Engset.

1. Offered and Carried Load

The parameter $a = \lambda/\mu = \lambda\tau$ defined in Eq. (13) is called the *offered load* and is measured in dimensionless units called *erlangs*. The offered load, which is a measure of the demand on the system, equals the mean number of arrivals per holding time. Equation (14) shows that the state probabilities P_j depend on the arrival rate λ and the mean holding time τ only through their product a; that is, the demand is completely specified by the number of erlangs.

According to PASTA Eq. (5), the Erlang B formula gives both the fraction of time the system will be in the blocking state and the fraction of calls that will be lost (because they arrive when the system is in the blocking state) as a function of the offered load a and the number of trunks s; any two of these values uniquely determines the third. A family of graphs of the Erlang B formula is given in Fig. 1.

The *carried load* (in erlangs) a' is defined as the mean number of busy trunks. When blocked calls are cleared, then $a' \equiv a'_{BCC}$ is given by

$$a'_{BCC} = \sum_{j \leq s} jP_j \tag{16}$$

Substitution of Eq. (14) into Eq. (16) gives, after some easy algebra.

$$a'_{BCC} = a[1 - B(s, a)] \tag{17}$$

Equation (17) can be interpreted to say that the carried load equals the product of the offered load a and the fraction $1 - B(s, a)$ of the offered load that is not lost; that is, carried load equals offered load minus lost load:

$$a'_{BCC} = a - aB(s, a) \tag{18}$$

If we imagine that $s = \infty$, then clearly $B(\infty, a) = 0$, and Eq. (18) shows that then $a'_{BCC} = a$. This provides another interpretation of offered load: offered load equals the mean number of busy trunks (that is, the mean number of simultaneous calls in service) in a system in which no calls are lost. (This interpretation is not restricted to BCC systems, as discussed below.)

The unit of offered load, defined in Eq. (13), and carried load, defined in Eq. (16), is the erlang, a dimensionless quantity. According to Eq. (16), the carried load (in erlangs) is the mean number of simultaneous calls in progress; from Eq. (18), the offered load is the mean number of simultaneous calls that *would be* in progress if the number of trunks were infinite; the lost load is the difference between them. In traditional telephony, loads are often measured in units called CCS, which stands for hundred-call-seconds per hour. This convention is based on technology; carried loads are measured by sampling the state of a trunk every 100 seconds for 1 hour and recording the number of times the trunk is found to be busy. Thus, if a trunk were busy continually throughout the hour, its carried load (1 erlang) would be recorded as 36 CCS (because there are 3600 seconds in an hour). Hence, 1 erlang of traffic equals 36 CCS (the "per hour" is usually not stated explicitly). The load carried by the trunk group is the sum of the loads carried by each trunk. Obviously, the use of CCS as the unit of traffic is highly arbitrary, and it is not used outside telephony. All of the formulas given here require that the loads be expressed in erlangs.

In telephony, the total time during which all trunks in a group are simultaneously busy is called ATB (All Trunks Busy), the number of calls that arrive is called PC (Peg Count), and the number of calls that are blocked is called O (Overflow). Then, the probability of blocking is estimated by ATB (per hour) and O/PC; if the arrival process is Poisson, then these measurements would be, in principle, equal over the long run. Which measurement is a better estimator of loss is a complicated statistical question, part of the subject of traffic measurement (discussed in a separate section).

If we define the system *utilization* ρ to be the carried load per trunk, then

$$\rho_{BCC} = \frac{a[1 - B(s, a)]}{s} \tag{19}$$

If the trunks are numbered 1, 2, ... and each arriving call is carried by the lowest-numbered idle trunk (*ordered hunt*, *ordered entry*), then the load carried by (on) the jth

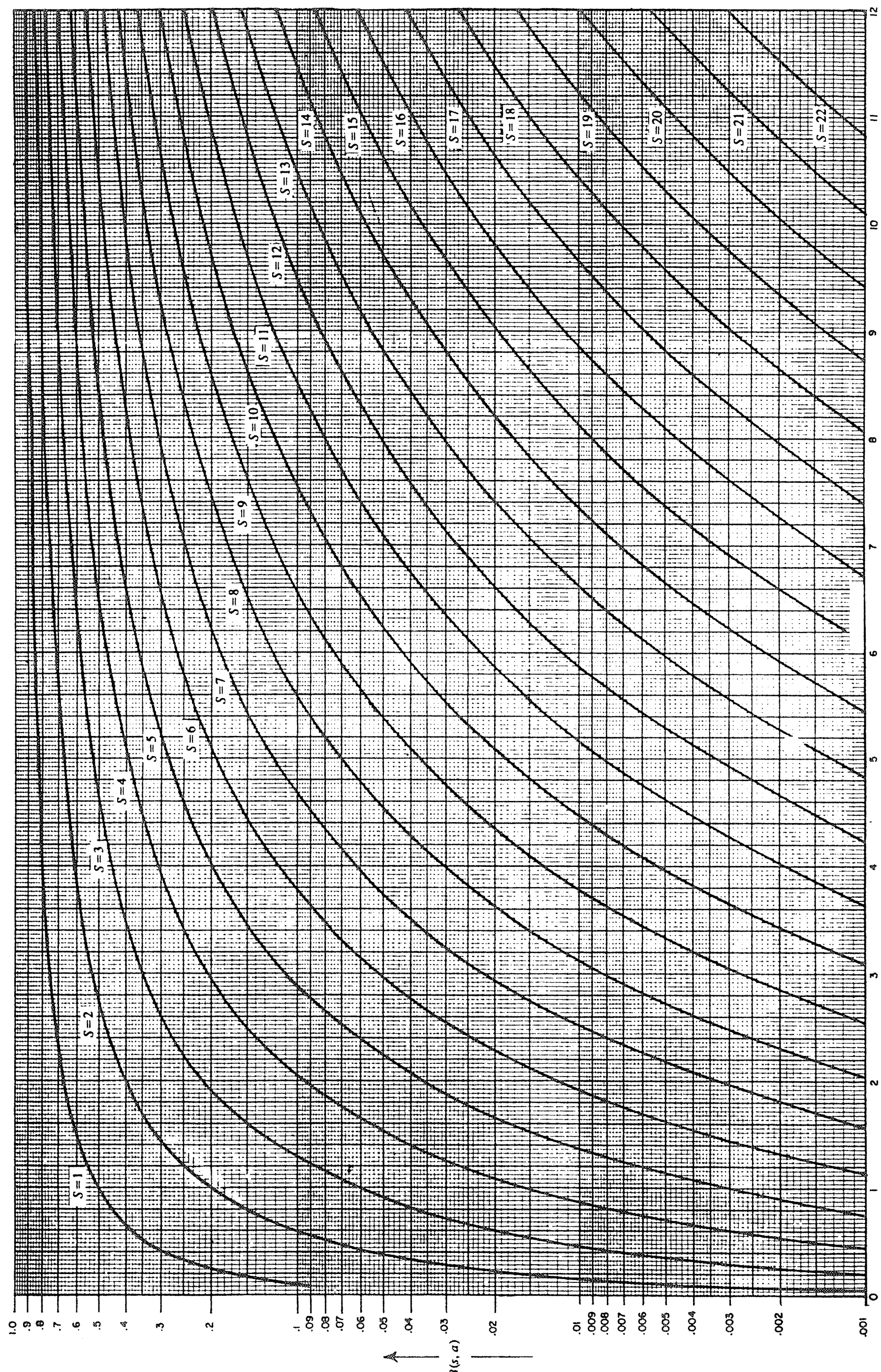

Figure 1 Graphs of the Erlang B formula.

trunk is the difference between the load $aB(j-1, a)$ that overflows trunk $j-1$ and the load $aB(j, a)$ that overflows trunk j:

$$a'_{BCC}(j) = aB(j-1, a) - aB(j, a) \tag{20}$$

where, of course, $B(0, a) = 1$. Also, $a'_{BCC}(j)$ equals the utilization of the jth trunk, that is, the fraction of time that trunk j is busy (but not the fraction of overflow calls from trunk $j-1$ that find trunk j busy, because overflow traffic is not Poisson, and PASTA does not apply). Note that, of course,

$$a'_{BCC} = \sum_j a'_{BCC}(j) \tag{21}$$

which follows from Eqs. (18) and (20).

It is difficult to calculate numerical values of the Erlang B formula directly from Eq. (15) when a or s are large. But, it is easy to show that

$$B(n, a) = \frac{aB(n-1, a)}{n + aB(n-1, a)} \qquad (n = 1, 2, \ldots, s;\ B(0,\ a) = 1) \tag{22}$$

and to write a computer program that implements Eq. (22). This algorithm is very fast and stable.

2. Insensitivity

Although the assumption of exponential holding times was tacitly used in our application of Eq. (1), it turns out that this assumption is not necessary for the conclusion Eq. (14) [and Eq. (15)] to be valid. Amazingly, when the blocked calls are cleared (and, of course, when $s = \infty$), the birth-and-death equations remain valid; the state probabilities for the Erlang loss system are *insensitive* to the form of the holding time distribution (that is, the holding times affect the state probabilities only through their mean value). Obviously, the study of insensitivity in stochastic systems is of great mathematical interest and practical importance.

A consequence of this insensitivity property for the Erlang loss system is the following theorem: If two independent Poisson streams of traffic, say a_1 erlangs and a_2 erlangs, are offered to a group of s trunks and blocked calls are cleared, then each stream sees the same probability of blocking, and it is given by the Erlang B formula $B(s, a)$ with $a = a_1 + a_2$, even if $\mu_1 \neq \mu_2$ and the holding times of the calls from each stream have different distributions. (Clearly , this theorem generalizes to an arbitrary number of independent Poisson streams.)

3. Efficiency of Large Trunk Groups

Numerical investigation [via Eq. (22), for example] of the Erlang B formula shows that large trunk groups are more efficient than small ones. For example, $B(1, 0.8) = 0.4444$, $B(10, 8) = 0.1217$, $B(100, 80) = 0.003992$, and $B(1000, 800) = 10^{-12}$. Likewise, $B(s_1 + s_2, a_1 + a_2) < B(s_1, a_1) + B(s_2, a_2)$.

The Erlang B is the most important and fundamental model in traditional teletraffic engineering. In modern wireless systems, where traffic is generated by mobile subscribers via cell phones in cars, the Erlang B remains a good model for the provision of radios in cell sites. This is true because, despite the mobility of the subscribers generating the calls, the assumptions (1)–(3) for the Poisson arrival process are still met; and the insensitivity of Eq. (15) to the distribution of holding times means that the truncation of holding times caused by handovers does not negate its validity for describing the effects of mobile traffic in which the blocked calls are cleared.

4. Simulation

It is instructive to study the simple simulation of a loss system, coded in simple BASIC, in Table 1. The code implements the ordered hunt procedure for assigning calls to trunks: the trunks are numbered $J = 1, 2, \ldots$; each arriving call is assigned to the lowest-numbered idle trunk; and the blocked calls are cleared. Instructions 140 and 200 specify the distribution functions of the interarrival times and the holding times, respectively. For example, using the *inverse transform method* (see any text on simulation), an exponential random variable realization with mean value M is generated whenever the value – M*LOG(1 − RND) is computed (where RND is a computer-generated random number). Thus, the program will simulate a loss system with Poisson arrivals with rate L and constant holding times with value T if the instructions are

```
140 IA = −(1/L)*LOG(1 − RND)
200 X  = T
```

It is instructive to run this simulation with different distributions specified by 140 and 200 (but with a fixed given offered load L*T), to compare the resulting values of

Table 1 Simulation of Loss System

No.	Instruction	Explanation
100	DIM C(50)	50 is the maximum number of trunks
110	INPUT S,N	S,N = number of trunks, calls to be simulated
120	NC = NC + 1	NC = number of calls simulated so far
130	J = 0	
140	IA =	IA = interarrival time
150	A = A + IA	A = arrival time
160	J = J + 1	J = index of trunk being probed
170	IF J = S + 1 THEN K = K + 1	K = number of calls that are blocked
180	IF J = S + 1 THEN 280	
190	IF A < C (J) THEN 160	C(J) = completion time for trunk J
200	X =	X = holding time
210	SX = SX + X	SX = sum of holding times for carried calls
220	C(J) = A + X	
230	M = C(1)	M = shortest trunk-completion time
240	FOR 1 = 2 TO S	
250	IF C(1) < M THEN M = C(I)	
260	NEXT I	
270	IF M > A THEN AB = AB + M-A	AB = cumulative time during which all trunks are busy
280	IF NC < N THEN 120	
290	PRINT K/NC, AB/A, SX/A	Fraction of calls blocked, fraction of time all trunks simultaneously busy, carried load

K/NC (the fraction of calls that are blocked) and AB/A (the fraction of time spent in the blocking state), and to compare these experimental values with the predictions of teletraffic theory (such as Erlang B values, PASTA, and insensitivity). It is easy to augment this code to include measurement of other quantities (or to allow the blocked calls to wait in a queue, etc.).

B. Erlang C Model

We now discuss briefly some of the other basic teletraffic models. If (1) the calls arrive according to a Poisson Process (as in Erlang B), (2) the holding times are exponentially distributed (not required for Erlang B), and (3) the blocked calls wait in a queue until a trunk becomes available (*Blocked Calls Delayed* [BCD], different from Erlang B), then the state probabilities are determined by the birth-and-death Eq. (1) with $\lambda_j = \lambda$ (Poisson arrivals) and

$$\mu_j = \begin{cases} j_\mu & (j \leq s) \\ s_\mu & (j > s) \end{cases}$$

(exponential holding times). Using these values for λ_j and μ_j in the birth-and-death equations yields, from Eq. (3)

$$P_j = \begin{cases} \dfrac{a^j}{j!} P_0 & (j = 1, 2, \ldots, s-1) \\ \dfrac{a^s}{s! s^{j-s}} P_0 & (j = s, s+1, \ldots) \end{cases} \tag{23}$$

and, if the infinite series in the denominator of Eq. (4) converges (that is, if $\lambda/s\mu = a/s < 1$), then

$$P_0 = \frac{1}{\displaystyle\sum_{k=0}^{s-1} \frac{a^k}{k!} + \frac{a^s}{s!(1-a/s)}} \qquad (a < s) \tag{24}$$

Then, the probability of blocking (the fraction of time that all trunks are simultaneously busy, which, by PASTA, equals the fraction of arriving calls that find all s trunks busy) is given by $P_s + P_{s+1} + \ldots \equiv C(s, a)$, the well-known *Erlang C* or *Erlang delay formula*:

$$C(s, a) = \frac{\dfrac{a^s}{s!(1-a/s)}}{\displaystyle\sum_{k=0}^{s-1} \frac{a^k}{k!} + \frac{a^s}{s!(1-a/s)}} \qquad (a < s) \tag{25}$$

If $a \geq s$, in which case the infinite series in the denominator of Eq. (4) diverges, then $P_0 = 0$ and, by Eq. (23), $P_j = 0$ for all finite j. Physically, the condition $a \geq s$ or, equivalently, $\lambda \geq s\mu$ means that the calls are arriving faster than the system can serve them in the long run, so we define $C(s, a) = 1$ when $a \geq s$.

Eq. (25) is analogous to Eq. (15); the Erlang C formula is sometimes called *Erlang's second formula*, denoted by $E_{2,s}(a)$, so $E_{2,s}(a) \equiv C(s, a)$. But, unlike the Erlang B formula, the Erlang C formula requires the assumption of exponential holding times; that is, the Erlang C model is *not* insensitive to the distribution of holding times.

The carried load a' [defined above as the mean number of busy trunks; see Eq. (16)] is given by

$$a'_{BCD} = \begin{cases} \sum_{j \le s} jP_j + \sum_{j > s} sP_j & (a < s) \\ s & (a \ge s) \end{cases} \tag{26}$$

Substituting Eqs. (23) and (24) into Eq. (26) yields

$$a'_{BCD} = \begin{cases} a & (a < s) \\ s & (a \ge s) \end{cases} \tag{27}$$

Equation (27) can be interpreted to say that, when $a < s$ in an Erlang C system, the carried load equals the offered load (because all arriving calls are carried). Simlarly, the system utilization ρ, defined above [see Eq. (19)] as the carried load per trunk, is given by

$$\rho_{BCD} = \begin{cases} \frac{a}{s} & (a < s) \\ 1 & (a \ge s) \end{cases} \tag{28}$$

As with the Erlang B formula, there is a better way to calculate the Erlang C formula than from its definition in Eq. (25); it is easy to show that

$$C(s, a) = \frac{sB(s, a)}{s - a[1 - B(s, a)]} \qquad (a < s) \tag{29}$$

It follows easily from Eq. (29) that $C(s, a) > B(s, a)$, which is explained by the observation that in the Erlang B system the blocked calls are cleared from the system, whereas in its Erlang C counterpart, the blocked calls remain in the system and thus can cause blocking for future arriving calls.

A family of graphs of the Erlang C formula is given in Fig. 2. For comparison with the numerical examples given above for the Erlang B formula, we give $C(1, 0.8) = 0.8$, $C(10, 8) = 0.4092$, $C(100, 80) = 0.01965$, and $C(1000, 800) = 5.6 \times 10^{-12}$. Note that in each of these examples, $\rho = 80\%$. This again demonstrates that large trunk groups are more efficient than small ones.

1. Waiting Times

In the Erlang C model, the blocked calls wait in the queue until a trunk becomes available. Let W be the waiting time of an arbitrary call. Then, for any order of service, $P(W > 0) = C(s, a)$. If the calls are served from the queue in the order of their arrival (FIFO; First In, First Out), then, it can be shown

$$P(W > t \mid W > 0) = e^{-(1-\rho)s\mu t} \qquad (t \ge 0) \tag{30}$$

that is, the waiting times for blocked calls are exponentially distributed, with mean value

$$E\ (W|W > 0) = \frac{1}{(1 - \rho)s}\tau \tag{31}$$

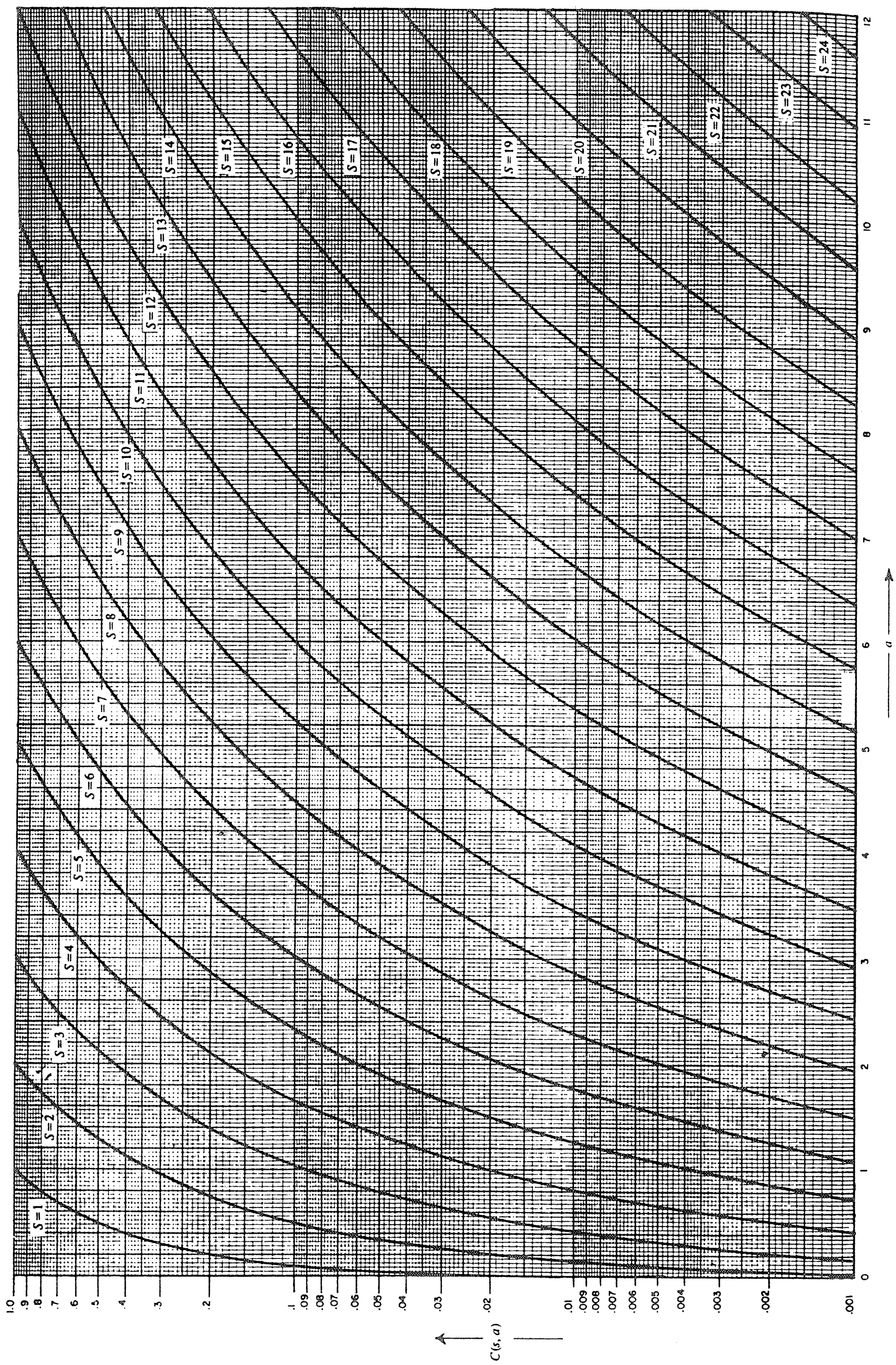

Figure 2 Graphs of the Erlang C formula.

Therefore, the unconditional (pertaining to *all* calls) waiting times are described by

$$P(W > t) = C(s, a)e^{-(1-\rho)s\mu t} \qquad (t \geq 0) \tag{32}$$

and

$$E(W) = \frac{C(s,\ a)}{(1-\rho)s}\tau \tag{33}$$

It is important to note that Eqs. (30) and (32) require that the service order be FIFO, whereas Eqs. (31) and (33) hold for all orders of service that do not depend on the service times of the calls in the queue.

All the formulas given above for the Erlang C model require that the holding times be exponentially distributed. However, for the special (but important) case when $s = 1$ trunk, we have the following well-known *Pollaczek-Khintchine formula*, which gives the mean waiting time for an arbitrary specification of the holding time distribution:

$$E(W) = \frac{\rho\tau}{2(1-\rho)}\left(1 + \frac{\sigma^2}{\tau^2}\right) \tag{34}$$

where σ^2 is the *variance* of the holding times. Equation (34) shows, for example, that the mean waiting time in the case of exponential holding times ($\sigma^2 = \tau^2$) is exactly twice as large as it is in the case of constant holding times ($\sigma^2 = 0$), all other things being held equal. Furthermore, for *any* distribution of holding times, when $s = 1$ Eq. (32) remains true for $t = 0$; that is,

$$P(W > 0) = \rho \tag{35}$$

Thus, in the single-server queue, the number of calls that are forced to wait is insensitive to the variability of the holding times [Eq. (35)], whereas the length of time that the blocked calls spend waiting in the queue is not insensitive, but instead depends on the amount of variability in the holding times [Eq. (34)]. (This phenomenon is a recurring theme throughout queueing theory.)

2. Effects of Retrials

We have already observed that $B(s,\ a) < C(s,\ a)$. The Erlang loss model does not account for the effect of blocked calls that retry. Clearly, the effect of retrials would be to increase the true probability of blocking beyond that predicted by the Erlang B formula. It is difficult to account precisely for the effect of retrials because the retrial stream does not follow a Poisson process (because it is not memoryless). One can take the viewpoint that, while the Erlang B formula underestimates the true probability of blocking (because it assumes that blocked calls never retry), the Erlang C formula overestimates the true probability of blocking (because it assumes that the blocked calls retry continually, with zero time between retrials, until they are served). An assumption that produces values that lie between these extremes is *blocked calls held*: every call spends its full holding time in the system whether or not it gets served. Then, the state probabilities are given by Eq. (1) with $\lambda_j = \lambda$ (Poisson arrivals) and $\mu_j = j\mu$ for all $j \geq 0$ (the aggregate call departure rate is the same whether $j \leq s$ or $j > s$); the solution is

$$P_j = \frac{a^j}{j!}e^{-a} \qquad (j = 0, 1, \ldots) \tag{36}$$

where, as before, $a = \lambda/\mu$ is the offered load. Equation (36) defines the *Poisson distribution* (not to be confused with the Poisson process). The probability of blocking is denoted by the *Poisson formula* $P(s, a)$:

$$P(s,a) = \sum_{j=s}^{\infty} \frac{a^j}{j!} e^{-a} \tag{37}$$

The model described by Eq. (37) seems very artificial, but it does produce "intermediate" values,

$$B(s,a) < P(s,a) < C(s,a)$$

The Poisson formula can be viewed as a way to account for retrials, to account for variation in the (assumed constant) arrival rate, or as a "fudge factor" to justify the provision of additional "safety" capacity beyond that indicated by the Erlang B formula. The Poisson formula is not used in modern teletraffic engineering, but we have included this discussion here because its existence in past practice often raises questions among engineers who do not know its history.

C. Engset Models

The Erlang B, Erlang C, and Poisson models all assume that the call arrival process is a Poisson process. A more general arrival process that still fits within the framework of the birth-and-death process is *quasirandom input*: the calls are generated by n independent, identical subscribers, each of which generates calls at rate γ when idle (and rate zero when waiting or in service). Then, when the system is in state j, the aggregate instantaneous call arrival rate is $(n - j)\gamma$; that is, take $\lambda_j = (n - j)\gamma$ in Eq. (1). Then, one can derive formulas analogous to the Erlang B and Erlang C (and Poisson) by making the corresponding assignments for the service completion rates μ_j. These "finite-source" models are often called Engset models after the author who first (1918) considered the finite-source analog of the Erlang loss model.

An interesting fact about models with quasirandom input is the arrival theorem: if $\Pi_j[n]$ and $P_j[n]$ are, respectively, the arriving customer's distribution and the outside observer's distribution for a birth-and-death model with n sources, then

$$\Pi_j[n] = P_j[n-1] \tag{38}$$

Equation (38) can be interpreted to say that the arriving customer sees the system as if he were an outside observer of the same system with himself removed from the calling population.

For example, when blocked calls are cleared, then the analog of Eq. (14) is (we assume $n > s$ to avoid trivialities)

$$P_j[n] = \frac{\binom{n}{j}\left(\frac{\gamma}{\mu}\right)^j}{\sum_{k=0}^{s}\binom{n}{k}\left(\frac{\gamma}{\mu}\right)^k} \tag{39}$$

and therefore the fraction of requests for service that are blocked is

$$\Pi_s[n] = \frac{\binom{n-1}{s}\left(\frac{\gamma}{\mu}\right)^s}{\sum_{k=0}^{s}\binom{n-1}{k}\left(\frac{\gamma}{\mu}\right)^k} \tag{40}$$

(including "retrials," since blocked sources remain eligible to generate new requests). This model has some interesting properties, but here we mention only that, like the Erlang loss model, the probabilities of Eqs. (39) [and (40)] are insensitive to the form of the holding time distribution. Moreover, these probabilities are insensitive to the distribution that governs the times between calls for each subscriber; all that is required is that the mean time between the instant a subscriber becomes idle and the next time the subscriber makes a request for service be $1/\gamma$ for each subscriber.

It is easy to show that, in the limit as $n \to \infty$ and $\gamma \to 0$ with the constraint $n\gamma = \lambda$, the quasirandom input (finite-source) models converge to their Poisson input (infinite-source) counterparts. [Thus, taking limits in Eq. (38) produces a result consistent with PASTA.] Finite-source models are more complicated than their infinite-source counterparts, so they are used only when the number of subscribers is relatively small and the ratio of subscribers to trunks is relatively low.

D. Some References

The discussion above gives the highlights of those aspects of queueing theory that are fundamental to classical teletraffic theory. Much of this material is covered in greater detail in Ref. 1, which is a queueing theory text with some emphasis on teletraffic models, and Ref. 2 which is a survey with an updated list of references. Reference 3 is a comprehensive and authoritative guide to the classical theory, especially as developed from the time of Erlang through the late 1950s. References 4–6 provide good treatments of background material in probability and stochastic processes, together with material that relates directly to queueing theory.

V. TRAFFIC MEASUREMENTS

The previous sections have described queueing models for which the parameters are known. In this section, we discuss some issues that arise in making measurements on operating traffic systems and in using these measurements for estimating parameters. Most of the theory and engineering practices in the United States were developed prior to the breakup of the Bell System in 1984 and were focused on voice communications. Since then, data traffic has become a larger proportion of the total traffic, and important characteristics of voice traffic have changed. We describe the classical traffic measurements and analyses in some detail and sketch some of the current issues that are motivating changes in the classical measurements.

A. The Classical Problem

The classical traffic measurement problem occurs in the setting of the Erlang B model. This is applicable to lines (circuits from customers to switches) and to trunks (circuits between switching systems) for traffic that is predominantly voice calls not overflowing from another network element. This typically justifies the assumption that calls arrive according to a Poisson process. The offered load is $a = \lambda/\mu$ as described in Eq. (13). There are s servers, blocked calls are cleared (BCC), and the system should block calls with probability no larger than b_o (b_o is called the *blocking objective* and typically equals 0.01). The assumption that the call attempts form a Poisson process is not required for the theory that follows. Some features of the formulas are negligible for Poisson traffic, but they may be negligible otherwise.

There is usually no difficulty in measuring s. The main issues concern measurements of the offered load a and the blocking probability b. Let $\hat{a}$ and $\hat{b}$ be the measured values of a and b, respectively; these measurements are called the *observed load* and the *observed blocking*, respectively. We want to know if $\hat{b} \le b_o$, and, if not, how many more servers are needed; $\hat{a}$ would be used to answer that question. In the context of the Erlang B model, these measurements are random variables, so we need to know something about their distributions. A consequence of the inherent randomness of measurements on a stochastic system is that $\hat{b}$ can differ from b_o even when they are theoretically equal. It is important to distinguish statistical fluctuations in $\hat{b}$ when $b \le b_o$ from a valid indication that $b > b_o$.

Among the decisions that have to be made are which observations to collect and over which time periods to collect them. We consider these questions in reverse order.

1. Engineering Periods

The queueing formulas that are the basis for traffic engineering assume that the arrival rate is not changing with time. The content of formulas such as the Erlang B and C formulas are steady-state, or long-run, probabilities. Experience has shown that call attempts vary with time of day. There is a tendency for peaks in the morning and afternoon due to business activity, and sometimes there is a peak in the evening from residential activity. Therefore, we want to use the longest interval in which the traffic parameters are constant, which is smaller than one day. Peaks typically last for one to two hours, so one hour has been taken as the standard measurement unit. Measurements are taken during peak periods (called *busy hours*) so that the *grade of service* (GoS) will be achieved throughout the day. There is little evidence of systematic day-to-day variation on standard workdays.

This means that busy hour measurements can be averaged over several days. In many geographical areas, there are periods during the year when the daily peaks are higher than normal; this is most obvious in resort areas. These are called *busy seasons*. Measurements are taken during the *busy season busy hour* (BSBH), typically one particular hour over five weekdays for four consecutive weeks; this is called the *engineering period*. The average of these 20 measurements is called the *average busy season busy hour* (ABSBH).

The BSBH is appropriate for measurements on network links because the GoS for links is often expressed as a blocking objective, and blocking probabilities are computed from the average load. Even for properly engineered links, congestion occurs when there are statistical fluctuations above the average load. Congested switching systems try to route some of their load to other switches, so switch congestion has the potential to spread. This means that peak loads are of more concern than average loads for components of a switching system. Engineering periods other than the ABSBH are used for these components. Some examples are the highest BSBH, the weekly peak hour (which may not be a BSBH), and the average of the 10 highest BSBHs.

In the United States, the days with the most long distance telephone traffic are Mother's Day and Christmas. Measurements taken on these days are used for designing and testing overload controls, not for capacity planning.

2. Measurements

Now we describe how $\hat{a}$ and $\hat{b}$ are measured and give some statistical properties of these measurements in the setting of the Erlang B model. The article by Hill and Neal is the source for these results (7). The measurement interval (BSBH) is denoted by $(0, T]$. The three measurements are

$$\begin{aligned} A(T) &= \text{the measured number of arrivals (peg count)} \\ O(T) &= \text{the measured number of overflows} \\ L(T) &= \text{the measured carried load} \end{aligned}$$

The carried load is the average number of busy servers, so if $S(t)$ is the number of busy servers at time t,

$$L(T) = \frac{1}{T}\int_0^T S(t)dt \tag{41}$$

Measurements are taken on n days (typically $n = 20$); a subscript i is used to denote day i.

We first discuss $\hat{a}$. The measured offered load on day i is the measured carried load divided by the proportion of the arrivals that are carried, so the measurements version of Eq. (18) is

$$\hat{a}_i = \frac{L_i(T)}{1 - \dfrac{O_i(T)}{A_i(T)}} \qquad (i = 1, 2, \ldots, n) \tag{42}$$

Let $\overline{a}$ be the average of the daily measurements, so

$$\overline{a} = \frac{1}{n}\sum_{i=1}^{n} \hat{a}_i \tag{43}$$

and let $Var(\hat{a})$ be the sample variance of the daily measurements, so

$$Var(\hat{a}) = \frac{1}{n-1}\sum_{i=1}^{n}(\hat{a}_i - \overline{a})^2$$

The variability in the daily measurements is attributable to day-to-day variation in the offered load and to finite sampling effects. An analysis of empirical data showed that a gamma distribution is a good fit to the distribution of observed loads. The gamma distribution has a density function denoted by $\gamma(\cdot)$, where

$$\gamma(x) = \frac{\beta(\beta x)^{\nu-1}}{\Gamma(\nu)} e^{-\beta x} \qquad (x \geq 0)$$

where Γ is the gamma function defined as

$$\Gamma(\nu) = \int_0^\infty t^{\nu-1} e^{-t} dt$$

When ν is a positive integer, $\Gamma(\nu) = (\nu - 1)!$ and the parameters ν and β are called the *shape* and *scale* parameters, respectively, and are nonnegative. The mean and variance of this distribution are ν/β and ν/β^2, respectively, so the mean and variance determine the distribution. The data also showed that the variance of the measured offered load is related to the mean via

$$Var(\hat{a}) = 0.13\overline{a}^{\phi} \tag{44}$$

where ϕ is a parameter that describes the amount of day-to-day variation. Three values of ϕ (1.5, 1.7, and 1.84) were chosen to describe low, medium, and high day-to-day variation, respectively.

Probabilistic analysis yields

$$Var(\hat{a}) = Var(a) + \frac{2\overline{a}}{\mu T} \tag{45}$$

where $1/\mu$ is the mean call holding time. The first term on the right in Eq. (45) is the variance due to day-to-day variation, and the second term is an approximation for variance due to the finiteness of the measurement interval. Substituting Eq. (44) into Eq. (45), rearranging, and ensuring that $Var(a) \geq 0$ yields

$$Var(a) = \max\left[0, 0.13\overline{a}^{\phi} - \frac{2\overline{a}}{\mu T}\right] \tag{46}$$

Equations (43) and (46) and $E(a) = \overline{a}$ are used to obtain the parameters of the gamma distribution that describes the day-to-day variation in a.

Now we examine $\hat{b}$. The measured blocking on day i is the proportion of the offered load that overflows, so

$$\hat{b}_i = \frac{O_i(T)}{A_i(T)} \qquad (i = 1, 2, \ldots, n) \tag{47}$$

Comparing Eqs. (42) and (47) shows that $\hat{a}_i$ and $\hat{b}_i$ are correlated. Let b be the average of the daily measurements, so

$$\overline{b} = \frac{1}{n}\sum_{i=1}^{n} \hat{b}_i$$

it is the *observed GoS*. Since a large number of arrivals tends to cause a large number of overflows, $A_i(T)$ and $O_i(T)$ are positively correlated. This means that $E(\overline{b}) \neq B(s, a)$ even when a is known precisely. Some lengthy analysis yields the approximation

$$E(\overline{b}) \doteq \int_0^{\infty} \left[B(s,a) = \frac{B(s,a)a\mu T - Cov(O_1(T), A_1(T))}{(a\mu T)^2}\right]\gamma(a)da \tag{48}$$

The second term in the integrand is negative in the range of engineering interest, so ignoring it leads to overestimates of the observed blocking. The magnitude of this term is negligible for Poisson traffic and is significant when the peakedness exceeds two. The formula for the covariance term is intricate and can be found in the article by Neal and Kuczura (8). Numerical integration is tractable for evaluating Eq. (48). Engineering design tools use Eq. (48) to obtain a design that will achieve the blocking objective. The empirical content of Eq. (48) is that this procedure makes the observed GoS agree with the designed GoS.

B. The Effects of Internet Calls

The classical measurement and analysis methods were developed when almost all telephone traffic was voice communications. Measurements of call holding times were consistent with an underlying exponential distribution, and the mean holding time was about three minutes. These properties of call holding times changed on local access links when dialing the Internet became popular in about 1995.

Internet access for most users consists of a voice call from the subscriber's premises (using a modem) to an Internet service provider (ISP) and packet transmission from the ISP to the Internet. The mean length of these calls is roughly 25 minutes, and the proportion of traffic they represent is growing rapidly. This has the effects of increasing the mean length of voice connections on the access links and changing the distribution of holding times from exponential to a mixture of two exponentials (which is called a *hyperexponential* distribution). The insensitivity of the Erlang B formula implies that the effect on the distribution will not affect the calculation of the objective blocking as long as the effect on the mean is taken into account. However, there is an effect on the analysis of traffic measurements.

The measured carried load is defined as an integral in Eq. (41). In practice, measurements are taken periodically (100 seconds apart is typical, yielding CCS, which are usually translated to erlangs), and the integrand is approximated by a step function. Numerical experiments show that this provides nearly the same measurement values that continuous observations do. The analysis leading to Eq. (48) uses a different representation of $L(T)$ to make the analysis tractable. To describe it, let h_{ij} be the holding time of the jth call on the ith day. Then,

$$L_i(T) \doteq \frac{1}{T}\sum_{j=1}^{A_i(T)} h_{ij} \tag{49}$$

This approximation includes that part of the holding time that lasts beyond T of a call that arrives during the measurement interval and excludes the part of a call that is in $(0, T]$ from calls that arrive before the measurement interval begins. This approximation is accurate when these errors are both small or when they cancel each other out; it is unlikely to be accurate when the probability that a call that arrives before time zero lasts beyond T. When the mean holding time is 3 minutes and T is 1 hour, P {call lasts longer than T} is $e^{-20} = 2.06 \times 10^{-9}$, which is negligible. When the mean holding time is 30 minutes, P {call lasts longer than T} is $e^{-2} = 0.135$, which is significant. As Internet traffic increases, the validity of Eq. (49), and therefore of Eq. (48), becomes more doubtful. The situation would be even worse if entertainment video over telephone lines becomes a popular service. The modifications to Eq. (48) that are required to mitigate the effects of these long holding time calls are not yet known.

VI. BROADBAND TRAFFIC

Digital voice links are capable of transporting 64,000 bits per second (64 kilobits per second [kb/s]). Broadband traffic refers to sources that transport at rates that are at least 24 times as large (North America and Japan), that is, at least 1.544 megabits per second (Mb/s), or at least 31 times as large (Europe), that is, at least 2.048 Mb/s. Three examples of broadband traffic are Internet traffic, videoconferences, and entertainment video over telephone lines. From a traffic engineering perspective, broadband traffic is qualitatively very different from voice traffic. The difference is caused by the way broadband traffic is carried on a telecommunications system.

Voice traffic is *circuit switched*, which means there is a path from the origin to the destination, with dedicated bandwidth that is established when the call is initiated and torn down when the call is terminated. The mean holding time for a subscriber is about 3 minutes, and the arrival rate is rarely more than 20 per hour per subscriber. Low-speed data (e.g., dial-up modems and facsimiles) are also circuit switched. When a circuit-switched call is accepted, it receives dedicated network resources. Consequently, there is no need to consider finer detail than the *call level*, that is, call arrival and holding times.

High-speed data are divided into packets of information, and these packets are the units of data transport; this is called *packet switching*. It is not uncommon for the holding time of a packet to be less than a millisecond and for thousands to arrive in a second. This quantitative difference becomes a qualitative difference in (at least) two ways. First, the interarrival times from a source are dependent. This occurs because, when a packet arrives, the source is transmitting data, so there is a greater chance (compared to independence) of another packet arriving soon. The operational manifestation of this is that packet arrivals are bunched, which is called *bursty traffic*. Second, buffers are provided to accommodate the temporary peaks in packet arrivals. Thus, at the packet level, we have a BCD system, whereas circuit switching is BCC. This means that waiting time and buffer overflow probabilities are the performance measures of primary interest at the packet level. At the call (or *connection*) level, which may be BCC, the usual circuit-switched performance measures apply.

New technologies and protocols are used to transport broadband traffic. The asynchronous transfer mode (ATM), Ethernet, and the Transmission Control Protocol/Internet Protocol (TCP/IP) suite are some of these. Detailed traffic models that describe the specific effects of each protocol are beyond the scope of this article; we describe some general ideas.

A. Packet Traffic

Figure 3 shows 30-second segments of four packet data traces. Each plot shows the arrival rate as a function of time. These traces are of traffic destined for the Internet, a videoconference, and two codings of movies. Notice that these traces do not resemble each other, so the application and the method of digital coding appear to affect the traffic characteristics significantly.

The Internet traffic data are the number of bits that arrive in 100 milliseconds (ms) multiplied by 10 to give megabits per second. It alternates rapidly between high and low rates, occasionally reaching zero when no packets are sent. Motion pictures are a sequence of still pictures called *frames*. The data for the three video traces are the number of bits in each frame, scaled to give megabits per second. The details of the Motion Picture Experts Group (MPEG) coding scheme are not important here, except to note that there is a periodic feature to them. The periodicity is a dominant feature of the trace. The videoconference and a film were coded with an H.261 algorithm, which is another coding scheme. The videoconference is people talking in front of a stationary camera. A movie is a sequence of scenes, in which a scene is characterized

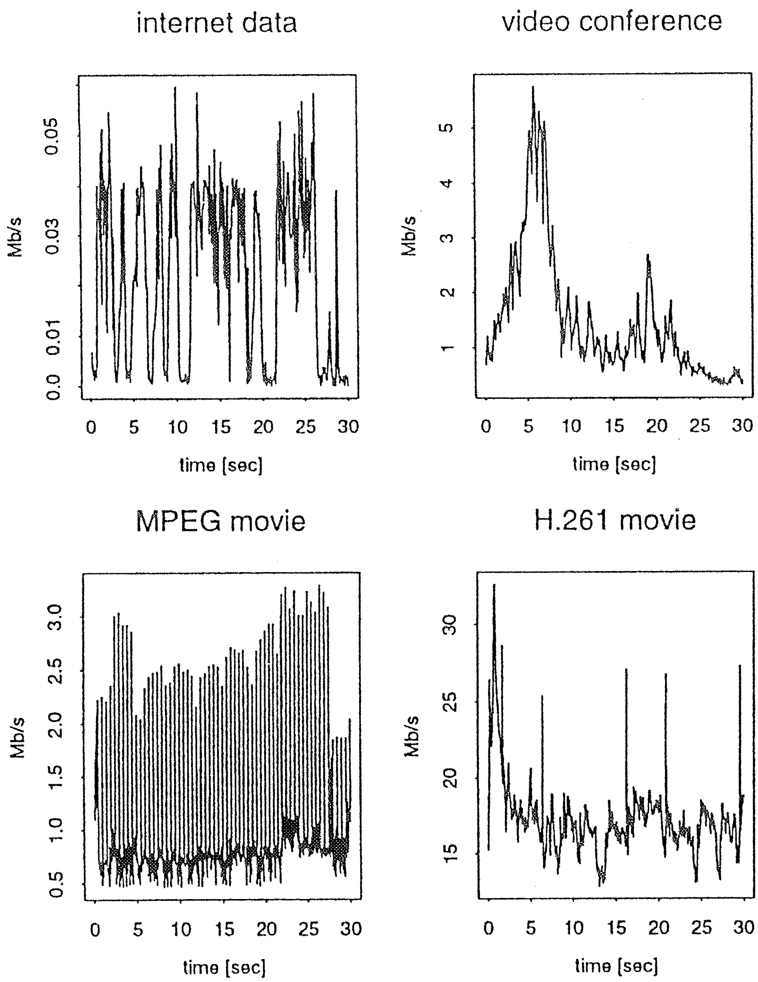

Figure 3 Broadband data traces shown as 30-second segments.

by a fixed camera angle and homogeneous subject matter. Video coding typically transmits the differences of adjacent frames, so scene changes require more bits than do frames within a scene. The film trace resembles a sequence of conferences separated by spikes, and the spikes represent scene changes.

Figure 4 investigates the videoconference trace in more detail. A Poisson process with the same rate was simulated. In the upper left plot, we see that the videoconference trace is much more volatile than the trace of the Poisson process. The other plots show the reason. The plots of the density functions for the number of bits per frame show that the tails of the Poisson distribution are not long enough; that is, there are too few observations significantly above and below the mean. The remaining plots show the relation between the number of bits in adjacent frames. For the videoconference, these points cluster closely around the 45° line, indicating strong correlation. (The correlation coefficient is 0.985.) For the Poisson process, the points appear to be uniformly spread, which is what we expect because the number of bits in adjacent

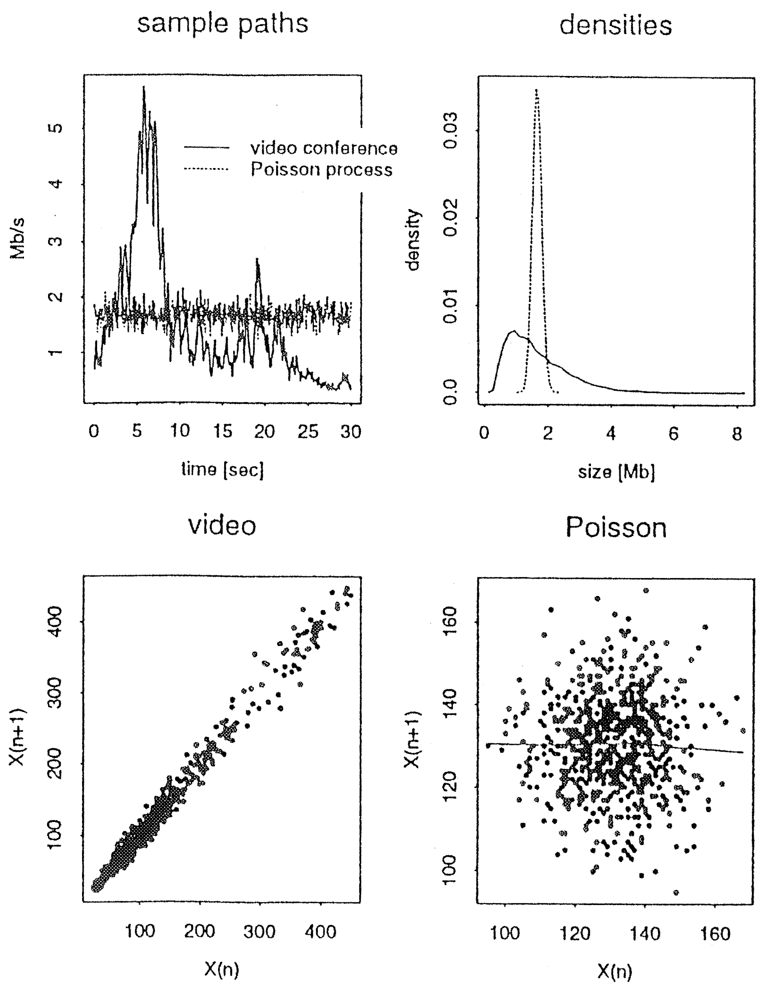

Figure 4 Time series, density, and correlation comparisons between a videoconference and a Poisson process.

frames is independent. The solid line in the plot is a nonlinear fit through the points; it is almost a horizontal straight line, indicating little empirical correlation.

Figure 4 shows that the Poisson process is not a good model for the videoconference trace. Similar analyses will show that it is not a good model for the other traces in Fig. 3. Several statistical models have been proposed for these traces, but none has achieved universal acceptance in the way the Poisson process has for first-offered voice traffic.

B. Effective Bandwidths

The lack of a statistical model for broadband traffic notwithstanding, engineers designing equipment to carry broadband traffic need a way to size buffers and other traffic-sensitive components to meet quality of service (QoS) requirements. Two problems that need to be solved are the planning problem and the connection admission control (CAC) problem. The former is to determine if a proposed system can carry a given traffic mix and satisfy the QoS constraints. The latter is to decide (in real time) if a new request for service should be accepted. A concept called the *effective bandwidth* of a source is being used by several manufacturers of ATM equipment for CAC and has been used for buffer sizing by at least one. A full treatment of this concept involves a considerable mathematical development, so only a simplified introduction is given here.

For analog signals, the *bandwidth* is the range of frequencies in the waveform. When all telephone signals were analog, the notion of bandwidth was used to describe the transmission requirements of a signal and the capacity of a transmission channel. A channel with capacity c could transport n_j signals of type j, each having bandwidth of w_j ($j = 1, 2, \ldots, j$) if, and only if,

$$\sum_{j=1}^{J} n_j w_j \leq c \tag{50}$$

was valid. This equation can be used to solve both of the problems just mentioned. For broadband traffic, we would like to find an analog of w_j that collapses the statistical properties of the traffic and its QoS requirements into a number that can be used in an equation similar to Eq. (50). When such a number is found, it is called the *effective bandwidth* (EBW) of the source. Notice that the EBW depends on the QoS, as well as the statistical properties, so the EBW of a source may be different in different applications. Hui first obtained Eq. (50) for a model of broadband traffic (9). Kelly surveys several EBW models (10).

There are several ways to model packet traffic and obtain EBWs. Capturing the bursty nature of packet traffic necessarily requires more complex stochastic processes than are used in the sections on Erlang B, Erlang C, and Engset models. These processes need more advanced mathematical methods to be brought to bear on the models. We present an EBW model that is widely applicable and relatively easy to explain (11). However, the explanation is more sophisticated than the explanations given for the Erlang and Engset models. We start with describing the model for a single source and extend the discussion to multiple sources.

A single source of packets is fed to a single server that works at rate c. There is a buffer of size B to handle temporary overloads. The QoS is the probability of buffer overflow. The single source is an aggregation of many sources that are producing packets, so the total number of packets behaves as if they flow in like a fluid. The flow rate varies as a Markov chain; this models the changes in the activity rates of the individual sources. Let m_{jk} be the rate (number of transitions per unit time) that the Markov chain makes transitions from state j to state k. When the chain is in state j, the flow rate is η_j. Let $M = (m_{ij})$ be the rate matrix of the Markov chain and ψ be the stationary distribution of M. Then, the mean arrival rate $\overline{\eta}$ equals $\Sigma \psi_j \eta_j$. To ensure stability, assume that the mean arrival rate is less than the service rate (i.e., that $\overline{\eta} < c$). The buffer will never overflow if the arrival rate never exceeds c, so assume, to avoid trivialities, that (at least) the largest arrival rate exceeds c. The following two examples illustrate the source model.

Example 1. Assume there are n subscribers, their connection attempts form a quasirandom input process, and successful attempts send packets at rate η for an exponentially distributed amount of time. Then, the number of active subscribers fluctuates as in an Engset model, and the state of the Markov chain is the number of active subscribers. Using the notation of the section on birth-and-death processes, $m_{j,j+1} = \lambda_j$ and $m_{j,j-1} = \mu_j$. The arrival rate in state j is given by $\eta_j = j\eta$.

Example 2. The videoconference data shown in Fig. 3 has been modeled as a Markov chain in the following way. Recall that a moving picture is a sequence of still pictures (called frames) that (ideally) arrive equally spaced in time. Let X_k be the number of bits in the kth frame; analysis of some videoconference data has shown that the sequence $\{X_1, X_2, \ldots\}$ is consistent with Markov chain behavior. We let the number of kilobits in the current frame be the state of the Markov chain, and m_{ij} is determined by the way the number of bits per frame changes. The arrival rate in state j is j kilobits per interframe arrival time.

Let $\Phi(B)$ be the steady-state probability that a buffer of size B overflows. The performance objective is to ensure that $\Phi(B) \leq p$, where p is typically small. To understand when the objective will be met, we consider what happens when

$$p \to 0 \qquad \text{and} \qquad B \to \infty \qquad \text{such that} \qquad \log p/B \to \zeta < 0 \tag{51}$$

the role of ζ is discussed below. It turns out that $\Phi(B)$ can be expressed in terms of the eigenvectors and eigenvalues of a certain matrix. As B gets large, the eigenvector-eigenvalue pair, for which the eigenvalue has the largest real part among all eigenvalues with a real part that is negative, basically determines $\Phi(B)$. That eigenvalue is related to $w(z)$, which is defined as the maximal real eigenvalue of the matrix

$$A(z) = H - \frac{1}{z}M$$

where H is the diagonal matrix, with $H_{jj} = \eta_j$, and z is a parameter. The following result can be proven with a substantial amount of analysis.

Proposition 1: Let Eq. (51) be valid. Then,

if $w(\zeta) < c$, then $\Phi(B) < p$

if $w(\zeta) > c$, then $\Phi(B) > p$

Because of this property, $w(\zeta)$ is the EBW of the aggregate source described by H and M.

This proposition is applied to a setting with service capacity c, buffer size B, and maximum allowable overflow probability p by setting $\zeta = \log p/B$, computing $w(\zeta)$ and comparing it to c. When the Markov chain has two states, $w(\zeta)$ can be written explicitly. Let

$$M = \begin{pmatrix} -\alpha & \alpha \\ \beta & -\beta \end{pmatrix}$$

α and β are the inverses of the expected lengths of times in States 1 and 2, respectively. Then,

$$w(\zeta) = \frac{(\eta_1 + \eta_2)\zeta + \alpha + \beta + \sqrt{[(\eta_1 + \eta_2)\zeta + \alpha + \beta]^2 - 4(\eta_1\eta_2\zeta^2 + \beta\eta_1\zeta + \alpha\eta_2\zeta)}}{2\zeta}$$

Since ζ depends on B, the EBW implicitly depends on the buffer size. This permits Eq. (50) to be used for buffer sizing.

Proposition 1 has been extended to cover multiple inhomogeneous sources. There are K sources as above, each with a transition matrix M_k and a diagonal matrix of flow rates $H_k, k = 1, 2, \ldots, K$. Let $w_k(z)$ be the maximal real eigenvalue of the matrix

$$A_k(z) = H_k - \frac{1}{z} M_k \qquad (k = 1, 2, \ldots, K)$$

Proposition 2: Let Eq. (51) be valid. Then,

$$\text{if } \sum_{k=1}^{K} w_k(\zeta) < c, \text{ then } \Phi(B) < p$$
$$\text{if } \sum_{k=1}^{K} w_k(\zeta) > c, \text{ then } \Phi(B) > p$$

A particularly interesting special case of Proposition 2 is when there are n_j sources of type j, $j = 1, 2, \ldots, J$. Then,

$$\text{if } \sum_{j=1}^{J} n_j w_j(\zeta) < c, \text{ then } \Phi(B) < p$$
$$\text{if } \sum_{j=1}^{J} n_j w_j(\zeta) > c, \text{ then } \Phi(B) > p$$

The traffic mix described by $(n_1, n_2, \ldots, n_j)$ sources of each type will violate the GoS criterion when the sum of the EBWs is larger than c and will satisfy it when the sum of the EBWs is less than c, which (except for the case of equality) solves the admission control problem and the planning problem.

There are some features of this model that are not as restrictive as a first glance might make them seem. The proven mathematical results are limit theorems as $p \to 0$ and $B \to \infty$; simulation experiments show that Propositions 1 and 2 are good approximations for a wide variety of realistic loss probabilities and buffer sizes. The sojourn time in any state of a continuous-time Markov chain has an exponential distribution; this may not conform to the sources at hand. However, by using more states, the sum or mixture of (possibly) different exponential distributions can be formed, and then sums and mixtures of these can be formed, and so forth. In fact, the Markov chain can be chosen to approximate a leaky bucket traffic shaper (which is described below).

C. Traffic Shaping and Policing

Some broadband services do not come with traffic limitations or performance guarantees. The Internet is an example. Sources may send packets as fast as they are able. The network tries to deliver the packets as soon as possible, but it does not promise to satisfy delay or packet-loss requirements. For other services (e.g., frame relay), the source and the network must agree on some characteristics of the offered traffic and the level of service that will be provided. The latter typically includes a minimum throughput, a maximum loss rate, and a bound on the maximum delay. The former typically includes an upper bound on the average arrival rate and the peak arrival rate. The obligations are expressed in a service-level agreement. If the source were to violate the service-level agreement, the network resources might be overloaded, causing the nonconforming source (and perhaps other sources) to receive substandard performance. To prevent this, the network employs a *policing* function to ensure compliance with the service agreement. Similarly, a source may want to ensure conformance to avoid service degradation, so it may shape its traffic to comply with the policing function. The *leaky-bucket algorithm* is a popular policing and traffic-shaping device; it is the subject of this section.

The leaky-bucket algorithm is very similar to a *token bank* or *credit manager* algorithm; the last two algorithms operate identically. The differences are slight enough to ignore here; our description is precise for the token bank and credit manager algorithms. How the policing algorithm works is discussed next.

Each user starts with an account of $C_{\max}$ tokens. This account is reduced by one token every time an *information unit* (think of this as a packet) is sent to the network. If a packet arrives when the account is zero, that packet is discarded. (Some services, e.g., frame relay, tag the packet as "discard eligible" and carry it when possible.) Periodically, at times Δ, 2Δ, 3Δ, ... , say, a new token is placed in the account, but this is suspended when the account contains C_{max} tokens. Thus, the long-run sustainable throughput is $1/\Delta$ packets per unit time, and C_{max} is the largest burst of packets that can be handled without loss.

Subscribers know the values of C_{max} and Δ, so they can implement a sending algorithm that conforms to the policing algorithm. They can emulate the algorithm, so they can always know the number of tokens in their account. When the account is zero, they can defer sending packets to the network. One way of doing this is to provide a buffer to hold the nonconforming packets. The buffer can be anyplace before the packets are interrogated by the policing algorithm. Typically, they are placed at the source, but some networks offer to provide them.

1. A Model for the Token Bank

Let $C(t)$ be the account balance at time t. The realizations of $C(t)$ are step functions that start at C_{max}, decrease by one when a packet arrives, and increase by one (as long they do not exceed C_{max}) at times $\Delta, 2\Delta, \cdots$. Let $C_n = C(n\Delta + 0)$ (i.e., right after time $n\Delta$), and let A_n be the number of packets that arrive in the interval $(n\Delta, (n + 1)\Delta)$. When there is no buffer (as in policing)

$$C_{n+1} = \min[(C_n - A_n)^+ + 1, C_{max}] \qquad (n = 0, 1, 2, \ldots) \tag{52}$$

When A_1, A_2, ... are independent and identically distributed, Eq. (52) describes a Markov chain. When $P\{A_0 = 0\} > 0$ and $P\{A_0 = k\} > 0$ for some $k > 0$, the Markov chain is aperiodic and irreducible, so it has a steady-state distribution. This distribution can be used to obtain the throughput, discard probability, and other performance characteristics.

There are several interesting ways to choose the distribution of the number of arrivals, but room does not permit exploring them here. The interested reader should consult Ref. 12.

Equation (52) can be modified to include a buffer of size B. It can be shown that adding the buffer to Eq. (52) is equivalent to changing C_{max} to $C_{max} + B$ and interpreting $C(t)$ as the number of tokens plus the buffer content at time t. The significance of this result is the following. Those packets that arrive when the token bank is empty are lost, so any pair of token supply and buffer positions that have the same sum have the same lost packets. Thus, tokens and buffer positions are interchangeable as far as the packet loss rate is concerned. They are not interchangeable in their effect on the shape of the output of the leaky bucket. For example, when there is one token and $B - 1$ buffer positions, the spacing between the outputs is Δ as long as the buffer is not empty. When there are B tokens and no buffer positions, as many as B packets can arrive and depart during an interval of length Δ as long as they arrive in such a way that the token is available at all arrival epochs (equispaced arrivals will do). The former scheme makes the output traffic smoother than the latter scheme does.

REFERENCES

1. Cooper, R. B. (1981). *Introduction to Queueing Theory*. 2nd ed. New York: North-Holland (Elsevier). Reprint (and Solutions Manual) available from University Microfilms International, Ann Arbor, MI.
2. Cooper, R. B. (1990). Queueing theory. In: Heyman, D. P., Sobel, M. J., eds. *Stochastic Models*. Amsterdam: North-Holland (Elsevier), pp. 469–518.
3. Syski, R. (1986). *Introduction to Congestion Theory in Telephone Systems*. 2nd ed. New York: Elsevier.
4. Heyman, D. P., Sobel, M. J. (1982). *Stochastic Models in Operations Research*. Vol. 1. In: Stochastic Processes and Operating Characteristics. New York: McGraw-Hill.
5. Ross, S. M. (1993). *Introduction to Probability Models*. 5th ed. San Diego, CA: Academic Press.
6. Wolff, R. W. (1989). *Stochastic Modeling and the Theory of Queues*. Englewood Cliffs, NJ: Prentice-Hall.
7. Hill, D. W. and Neal, S. R. (1976). Traffic capacity of a probability-engineered trunk group. *Bell Sys. Tech. J.* 55:831–842.
8. Neal, S. R. and Kuczura, A. (1973). A theory of traffic-measurement errors for loss systems with renewal input. *Bell Sys. Tech. J.* 52:967–990.
9. Hui, J. Y. (1988). Resource allocation for broadband networks, *IEEE J. Sel. Areas Commun.* 6:1598–1608.
10. Kelly, F. P. (1996). Notes on effective bandwidths. In: Kelly, F. P., Zachary, S., Ziedins, I., eds. *Stochastic Networks. Theory and Applications*. Oxford, UK: Clarendon Press.
11. Elwalid, A. I., Mitra, D. (1993). Effective bandwidth of general markovian traffic sources and admission control of high speed networks. *IEEE/ACM Trans. Networking*, 1:329–343.
12. Heyman, D. P. (1992). A performance model of the credit manager algorithm. *Computer Networks and ISDN Systems* 24:81–91.